**Part 1:**

$FF in hexadecimal, which is 11111111 in binary gets loaded into accumulator A. Then accumulator A stores $FF into data direction register port H to configure it into an output. PTT is loaded into accumulator A in order to read port T. The changes on PTT are stored into port H which gets looped infinitely via “BRA Loop”. When there are changes done to any of the switches that are loaded onto port T are then reflected onto port H where the output of the SEG of LED1 is displayed. If the switch is on, the light is on, if the switch is flipped off, the light is turned off.

**Part 2:**

BSET DDRP, %11111111 configures all of Port P into an output, which acts as the LED2 control). BSET DDRE, %00010000 turns the pin PE4 into an output. BCLR PORTE %00010000 and the not(OE), OE = output enable), turns it into 1 to enable the keypad. A loop is created where PTS reads a key code that is then loaded into accumulator A. LSRA appears 4 times in the loop which shifts right accumulator A. STAA PTP outputs the accumulator A content to LED2 which should display the light and then the loop will loop infinitely. The keypad should be reading the binary equivalent of the ASCII character on the keypad and translates it into the equivalent LED light. The LED only displays colours in RGB and its subset CMNY.

**Part 3:**

LDAA #%10000000 enables PP7, buzzer. EORA #%10000000 is a XOR which disables the buzzer. This creates a set of intervals where the buzzer is on and buzzes for a set duration and then is off for a set duration. This creates the frequency of the sound that is heard from the buzzer.